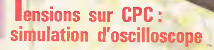
Micro Informatique CPC • PCW • PC1512



Jeu de dames sur Amstrad-Intosh

PAO : c'est possible !



Mensuel n° 30 - Janvier 1988

M 1355 - 30 20,00 F
3791355020009 00300

EDITO

Et voilà ! Encore une année de passée en votre compagnie !... cela fait bientôt trois ans que cela dure et c'est toujours avec le même enthousiasme que nous ouvrons cette nouvelle année en vous remerciant de votre fidélité.

Dans ce numéro, il y a, comme d'habitude un sujet d'intérêt pour chacun : des listings bien entendu mais aussi des articles didactiques accessibles aux débutants. Il n'est bien sûr pas question d'abandonner les sujets techniques qui nous valent toujours des éloges, mais ces derniers seront désormais plus commentés.

Il ne nous reste plus qu'à vous souhaiter d'une façon très traditionnelle :
BONNE ET HEUREUSE ANNEE 1988.

La rédaction

SOMMAIRE

Remerciements à Hewlett Packard France pour la photo de couverture.

LES CARACTERES DE CONTROLES

© Michel ARCHAMBAULT

Le courrier des lecteurs est souvent très instructif. C'est ainsi que l'on constate que vous êtes nombreux à ne pas avoir bien compris l'utilisation des CHR$ inférieurs à 32, appelés communément "caractères de Contrôles" ; et particulièrement parmi les possesseurs de 6128. Sans doute une question de clarté de notices...
Il est donc prudent de reprendre "à zéro" certaines notions indispensables ; puis nous ferons une vue d'ensemble de ces caractères, et enfin quelques applications pratiques ou spectaculaires.

CHR$ ET CTRL

PRINT et CHR$(65) veut dire affiche le "CHARACTER" (= caractère) dont le code ASCII est 65 ; ici le A majuscule. C'est donc rigoureusement identique à PRINT "A".
Or les codes ASCII qui correspondent à un caractère visible à l'écran vont de 32 (espace ou blanc) à 255 ; alors que de 0 à 31 ce sont des ACTIONS. Par exemple PRINT CHR$(12) efface l'écran, c'est donc identique à CLS.
Les codes de 32 à 127 font partie de la norme ASCII que tous les fabricants actuels respectent même AMSTRAD.
De 0 à 31 le respect de la norme n'est que partiel : les effets des codes 7, 10, 12 et 13 sont toujours les mêmes d'un micro à l'autre, pas pour les autres codes.
De 128 à 256 ce sont les "caractères graphiques". Alors là c'est l'anarchie totale ! Chaque marque a les siens. Même les fabricants d'imprimantes : je veux dire par là que si votre CPC pouvait envoyer ses codes sur huit bits (pour les supérieurs à 127), un CHR$(200) donnerait sur papier un tout autre motif que celui obtenu à l'écran.
Revenons à la norme ASCII tant bafouée des codes de 0 à 31 :
Il avait été prévu que l'on pourrait les obtenir directement au clavier par la touche CTRL + une lettre. Exemple CTRL L = CLS (code 12), CTRL G = BEEP (code 7), etc. Pas sur AMSTRAD CPC ! Si vous faites CTRL G sur un CPC vous n'obtenez pas un beep sonore, mais un petit caractère graphique représentant une sonnette. C'est le symbole graphique de CHR$(7). A présent tapez PRINT guillemet CTRL G guillemet, puis ENTER. Vous entendez un beep parce que le CPC a compris PRINT CHR$(7). Et si on mettait ces signes cabalistiques dans nos programmes BASIC ? Cela ferait trois caractères à taper (avec les deux guillemets") au lieu des sept de CHR$(7). DEFENDU ! En effet ce serait impossible à lister sur une imprimante, et voici pourquoi :
L'imprimante considère aussi les codes 0 à 31 comme des commandes, et lorsqu'elle recevra le code 7 de votre caractère obtenu par CTRL + G c'est elle qui fera BEEP ! Et en imprimant seulement
340 PRINT ""
En revanche elle pourra imprimer
340 PRINT CHR$(7)
parce qu'elle recevra des codes ASCII supérieurs à 31. (Le chiffre "7" a pour code 55)
Pire encore si pour programmer un CLS vous tapiez PRINT suivi d'un CTRL L entre guillemets ; car une imprimante recevant un code 12 fait un saut de page... Très amusant.
Donc en programmation défense de se servir de la touche CTRL, autrement qu'avec les touches flèches pour se déplacer plus vite dans une ligne de listing.

Avant d'aborder les codes de contrôles plus en détail rappelez-vous qu'ils doivent toujours être précédés par PRINT CHR$
IMPORTANT :
La norme ASCII a donné des noms de deux à trois lettres à ces commandes : SOH, STX, ETX, EOT, ENQ, etc. Ils figurent dans votre manuel ; n'en tenez pas compte car le BASIC du CPC ne connaît pas ces mots-là ! Ils n'existent même pas dans la liste des "Mots réservés".

LA LISTE DES CODES DE CONTROLES

Certains codes ont peu d'intérêt car ils ont un homologue sous forme de commande BASIC ; c'est plus facile à retenir qu'un nombre. D'autres sont utiles mais doivent être suivis d'un paramètre numérique, lui aussi entré sous forme de CHR$, nous verrons des exemples.
Nous n'allons pas les présenter dans l'ordre des numéros, car ce serait absurde sur le plan pédagogique, mais par genres d'actions et avec les plus utiles en tête.
Par "curseur" entendez la position courante d'écriture au moment du print CHR$.
Par "écran" sous-entendez "fenêtre" au cas où vous auriez créé des WINDOW.
NOTA : à cette liste nous pensons qu'il sera logique d'ajouter les codes ASCII compris entre 240 et 243. Ce ne sont pas des codes de contrôles mais ils ont des rôles très voisins.

Déplacements du Curseur Texte
13 - Retour à gauche de la ligne, ou = touche ENTER
11 - Remonte le curseur d'une ligne.
8 - Ramène le curseur d'un "cran" à gauche
2 - Suppression du curseur. (couleur fond)
3 - Fait réapparaître le curseur.
9 - Curseur à droite.





PRINT CHR$(1);CHR$(code ASCII)

0	□		10	↓	J	20	◷	T	30	◰	↑
1	┌	A	11	↑	K	21	⤫	U	31	◳	_
2	┴	B	12	⇓	L	22	⎍	V	32		
3	┘	C	13	↵	M	23	┤	W	33	!	
4	⚡	D	14	◶	N	24	⌛	X	34	"	
5	⊠	E	15	◵	O	25	♦	Y	35	#	
6	✓	F	16	⊟	P	26	⚲	Z	36	$	
7	Ω	G	17	◴	Q	27	⊖	[	37	%	
8	←	H	18	◷	R	28	◲	\	38	&	
9	→	I	19	◶	S	29	◱	]	39	'	

10 - Curseur vers le bas (saut de ligne)
30 - Equivalent d'un LOCATE 1,1.

Effacement de Texte

18 - Efface la partie de ligne à droite du curseur
17 - Efface la partie de ligne à gauche du curseur
16 - Efface le caractère à l'emplacement du curseur.
20 - Efface le bas de l'écran à partir du curseur
19 - Efface le haut de l'écran jusqu'au curseur.
12 - = CLS.

Effets de Couleurs

24 - Inverse PEN et PAPER pour ce qui suit (inversion vidéo)
22 - suivi de 1 = Mode transparent.
22 - suivi de 0 = supprime le mode transparent.
23 - suivi de 1, 2, 3 = effets de substitutions de couleurs.

Affichages spéciaux

5 - suivi d'un code ASCII : affiche ce caractère à l'emplacement du curseur graphique.
1 - suivi d'un code 1 à 31, affiche ce symbole graphique
21 - Stoppe l'affichage texte
6 - Rétablit l'affichage texte.

Divers

7 - Emet un BEEP sonore.
26 - Fin du fichier séquentiel (EOF)

Les inutiles

0 - sans effet
27 - sans effet à l'écran, mais très usité en PRINT #8.
4 - + paramètres = commande MODE
14 - + paramètres = commande PAPER.
15 - + paramètres = commande PEN.
25 - + paramètres = commande SYMBOL
26 - + paramètres = commande WINDOW.
28 - + paramètres = commande INK.
29 - + paramètres = commande BORDER
30 - + paramètres = commande LOCATE

Tout ceci n'était qu'un panorama rapide des codes de contrôles. Certaines descriptions succinctes vous laissent "dans le noir", alors que pour d'autres vous n'entrevoyez pas d'applications utiles. C'est donc le moment pour plonger dans le détail.

LES DEPLACEMENTS DE CURSEUR

Ces commandes sont pratiques pour des présentations sophistiquées de pages d'écrans. Ce très court exemple est assez "parlant".

```
10 CLS:LOCATE 10,10:PRINT "DEPART";
20 PRINT CHR$(11);"PLUS HAUT"
```

La CHR$(13) est souvent utilisé pour clore une série de PRINT suivis de points virgules.

```
10 FOR N=65 TO 75:PRINT CHR$(N);
20 NEXT:PRINT CHR$(13)
```

La présence du curseur peut être jugée disgracieuse lors de saisies de valeurs par INPUT. Très facile à supprimer

```
10 PRINT CHR$(2);:INPUT "NOM";N$
20 INPUT "PRENOM";P$
30 PRINT CHR$(3) : 'Retour à la normale
```

La codage des touches flèches : vous connaissez ces logiciels où l'on vous demande de choisir une option en vous déplaçant sur l'écran par les touches curseur ? Ceci fait le programme a enregistré ces coordonnées et en déduit votre choix. Mais savez-vous que ce n'est pas sorcier ?
Essayez d'abord.

```
FOR N=240 TO 243:PRINT CHR$(N);CHR$(32);:NEXT
```

Vous obtenez quatre caractères graphiques représentant des flèches, mais la plupart des "Amstradistes" ignorent que les "touches qui n'écrivent rien" ont leur code ASCII ! Tapez vite ces trois lignes

```
10 R$=INKEY$
20 IF R$<>"" THEN PRINT ASC(R$)
30 GOTO 10
```

Vous découvrez alors que les touches flèches, COPY, TAB, CLR, DEL et ENTER ont chacune leur code, mais à la condition essentielle de les interroger par INKEY$.

flèche haut = 240 ; flèche bas = 241 ; flèche gauche = 242 ; flèche droite = 243 , COPY = 224 , TAB = 81 ; CLR = 18 , DEL = 127 et bien sûr ENTER = 13

Tout ceci est mis en pratique par le petit programme-démo "ICI" (listing n° 1). Les lignes 110 et 120 vous empêchent de sortir de l'écran, tandis que la ligne 150 affiche en haut à droite les coordonnées du curseur.

Il vous sera alors facile d'adapter ce listing à vos besoins personnels.

Remarquez à cette occasion que ce programme fonctionne aussi avec les touches "normales" et leurs combinaisons (SHIFT ou CTRL + touche). Vous serez alors surpris de constater que l'on peut intercepter des SHIFT + flèche ou CTRL + flèche.

LES EFFACEMENTS DE TEXTE

Le plus fréquent d'emploi est CHR$(18) qui efface toute la partie droite de la ligne à partir du curseur. Il permet dans un INPUT d'effacer une réponse refusée. Exemple

```
10 CLS
20 LOCATE 10,12:PRINT CHR$(18);
INPUT "Un nombre > 1000 ",N8
30 IF N8<=1000 THEN PRINT
CHR$(7);:GOTO 20
```

Le libellé de la question est lui aussi effacé. On peut éviter cela par cette variante

```
20 LOCATE 10,12:PRINT"Un Nombre
> 1000 ",CHR$(18);:INPUT "",N8
```

CHR$(18) simule la touche CLR. Pour simuler la touche DEL il suffit de le faire précéder par un déplacement à gauche, le CHR$(8). Petite démo

```
10 LOCATE 1,3:PRINT STRING$(250,
"A");
20 FOR W=1 TO 2000:NEXT
30 FOR N=1 TO 250:PRINT CHR$(8);
CHR$(18);:NEXT
```

C'est une façon originale de présenter une suite de messages, en les effaçant par la fin après lecture.

En revanche, pour effacer depuis le début on fait suivre par un déplacement à droite, CHR$(9). Ajoutez la ligne

```
25 LOCATE 1,3
```

puis modifiez ainsi la ligne 30

```
30 FOR N=1 TO 250:PRINT CHR$(16);
CHR$(9);:NEXT
```

Ceci pour vous montrer qu'il ne faut pas se priver de combiner les codes de contrôles.

Une autre fantaisie d'affichage de phrases utilise CHR$(19) qui efface ce qui est au dessus, donc déjà lu. Petite démo

```
10 CLS:FOR C=65 TO 68
20 PRINT CHR$(19);STRING$(180,C)
30 FOR W=1 TO 2000:NEXT
40 NEXT
```

LES EFFETS DE COULEURS

Le plus courant est l'inversion vidéo pour mettre une chaîne en valeur ; c'est CHR$(24).

Attention c'est un "commutateur", puisqu'il inverse les INK de PEN et PAPER. Il ne faut donc pas oublier le retour aux teintes normales après usage ; autrement dit le total des CHR$(24) doit représenter un nombre pair.

Comme c'est une écriture à la longue fatiguante il est pratique de le désigner par une variable chaîne. Exemple

```
10 V$=CHR$(24)
20 PRINT "Il est";V$;"facile";V$;"de
faire l'inversion";V$;"vidéo";V$" "
```

La transparence permet de superposer des caractères sans que les seconds effacent les premiers. On peut ainsi souligner un mot, une phrase par le signe "_" (code ASCII 95), ou le barrer par des signes "-" ou "=". La transparence est mise en service par CHR$(22); CHR$(1) puis annulée par CHR$(22); CHR$(0).

N'oubliez pas de l'annuler après usage ! Ici encore on a intérêt à utiliser ces commandes sous formes de variables chaînes

```
10 TR$=CHR$(22)+CHR$(1):NTR$
=CHR$(22)+CHR$(0)
20 CLS
30 LOCATE 5,12:PRINT "SUPERPOSI-
TION PAR TRANSPARENCE"
40 LOCATE 5,12:PRINT TR$;STRING$
(13,95);NTR$
```

Dans le STRING$ remplacez 95 par 154. Ajoutez un PEN 2 en ligne 35 et PEN 1 en ligne 50. Toutes les fantaisies sont permises.

Un texte affiché en mode transparent n'efface pas non plus des traits graphiques obtenus par DRAW ou DRAWR. La preuve, insérez la ligne

```
37 PLOT 0,150,3:DRAWR 400,150
```

Attention, la transparence ne fonctionne pas si vous positionnez les chaînes par TAG au lieu de LOCATE.

L'effet de CHR$(23) ne concerne que le mode graphique : lorsqu'un trait en croise un autre vous avez déjà remarqué que la couleur au point d'intersection est celle du dernier trait (transparence ou non). Avec CHR$(23), suivi d'un paramètre entre 1 et 3, cette zone peut être d'une tout autre couleur. Disons tout de suite qu'elle est très complexe à prévoir car le raisonnement mathématique est des plus rebutants. Je préfère vous en faire grâce et vous présenter un petit programme de démonstration (listing n° 2) aux effets assez curieux.

AFFICHAGES SPECIAUX

CHR$(1) permet de s'enrichir de 32 caractères graphiques supplémentaires, ceux obtenus par PRINT CHR$(1);CHR$(C), C pouvant varier de 0 à 31 (et même jusqu'à 255 mais c'est sans intérêt). Le petit programme "CHR1" (listing n° 3) va vous les présenter proprement à l'écran, et ce avec le numéro de code

LISTING 1

```
10 ' ICI par Fleches - M.A. 10/87
20 MODE 1:BORDER 0:L=12:T=19
30 PEN 3:PRINT "  Touches fleches, puis
COPY":PEN 1
40 R$=INKEY$
50 IF R$="" THEN 140
60 LOCATE T,L:PRINT SPC(1)
70 IF R$=CHR$(240) THEN L=L-1
80 IF R$=CHR$(241) THEN L=L+1
90 IF R$=CHR$(242) THEN T=T-1
100 IF R$=CHR$(243) THEN T=T+1
110 IF L=25 THEN L=24 ELSE IF L=1 THEN L
=2
120 IF T=39 THEN T=38 ELSE IF T=0 THEN T
=1
130 IF R$=CHR$(224) THEN 170:'COPY
140 LOCATE T,L:PRINT CHR$(243)
150 LOCATE 34,1:PRINT USING"##";T;:PRINT
 USING":##";L
160 GOTO 40
170 BORDER 1
180 LOCATE 4,12:PRINT "Vous etiez en col
onne";T;"ligne";L:PRINT●
```



et la touche avec laquelle on peut les obtenir directement avec la touche CTRL. Exemple CTRL+G affiche une petite sonnette. Si vous disposez d'une routine de hard copy d'écran sur imprimante, je vous engage à ajouter à ce mini-programme une ligne qui lancera cette copie.
Rappelons que dans votre listing si vous programmez par CTRL+touche au lieu de PRINT CHR$(1),CHR$( ) vous ne pourrez pas lister sur imprimante.
CHR$(5) suivi d'un caractère ou de son code ASCII (en CHR$ bien sûr) affiche ce caractère à l'emplacement du curseur graphique. C'est une sorte de TAG mais pour un seul caractère ; il est souvent plus pratique d'emploi que ce dernier. Par exemple pour espacer plus les lettres d'un titre : le pas normal en MODE 1 est de 16 points graphiques, portons-le à 24.

```
10 MODE 1
20 A$="ESPACEMENT CHIC"
30 PLOT 30,200,1
40 FOR N=1 TO LEN(A$)
50 PRINT CHR$(5),MID$(A$,N,1);
60 MOVER 24,0
70 NEXT
```

LES CODES DIVERS

Le plus célèbre est le PRINT CHR$(7) qui émet un beep sonore, mais systématiquement pensez à le faire suivre d'un point-virgule. En effet il serait stupide que chaque beep d'alerte soit suivi d'un saut de ligne.
La touche ESC n'a pas de code, c'est une touche d'arrêt. A ne pas confondre avec d'autres micros ou la touche ESC a pour code 27.
Dans les AMSTRAD CPC le PRINT CHR$(27) n'a aucun effet parce que non programmé, tout comme CHR$(0). En revanche PRINT #8,CHR$(27); est destiné uniquement aux imprimantes : lorsqu'une imprimante reçoit le code 27 elle sait que ce qui suit n'est pas un caractère à imprimer mais une consigne d'écriture (= les codes "EPSON").
Exemple :
PRINT #8,CHR$(27),"E","HELLO"
imprimera HELLO en caractères gras. Sans ce #8 nous aurions à l'écran EHELLO.
CHR$(26) est le signal de fin de fichier, le fameux EOF. Il s'enregistre à votre insu lors du CLOSEOUT. Il peut être obtenu directement au clavier par CTRL+Z (symbole graphique un point d'interrogation à l'envers). Imaginons ce court programme de saisie de fichier :

```
10 OPENOUT "TEST-EOF"
20 FOR N=1 TO 6
30 INPUT "Nom",A$:PRINT #9,A$
40 NEXT
50 CLOSEOUT
```

Au troisième nom tapons CTRL+Z, ENTER (le bon gag...), puis les trois derniers noms. Tout s'est passé normalement. Oui mais à présent essayons de le relire :

```
60 OPENIN "TEST-EOF"
70 FOR N=1 TO 6
80 INPUT #9,A$:PRINT A$
90 NEXT
100 CLOSE IN
```

L'écran affiche les deux premiers noms puis plantage avec EOF met in 80 (= fin de fichier rencontrée en 80)

```
70 WHILE NOT EOF
90 WEND
```

Toujours les deux premiers noms seulement mais pas de plantage.
Conclusion un CHR$(26) "artificiel" provoque un CLOSEIN, les enregistrements suivants sont devenus inaccessibles. Le second CHR$(26), écrit par le CLOSEOUT, ne sera jamais atteint.
Nota pour les possesseurs de CPC 464 : ce phénomène explique le "EOF met" en faisant un MERGE ou CHAIN MERGE sur un programme BAS. C'est dû à la rencontre d'un octet égal à 28 en plein milieu d'une "ligne" (un programme BAS enregistré est partiellement codé en binaire).

CONCLUSION

La plupart des codes de contrôles constituent des commandes BASIC à part entière, qui complètent le répertoire classique. Par nos exemples nous avons montré que plusieurs d'entre elles sont super utiles et sans équivalents avec les "mots" BASIC. Le principal inconvénient est qu'il est plus difficile de se souvenir d'un nombre que d'un mot ; il est donc conseillé de les noter sur un feuillet que l'on collera (par exemple) dans le manuel.
Leur utilisation astucieuse (ou combinée) permette d'obtenir très facilement des effets spéciaux qui sortent de l'ordinaire.

LISTING 2

```
10 ' EFFET DE CHR$(23) - M.A. 10/87
20 MODE 0
30 FOR C=1 TO 3:L=C*3
40 PEN C:LOCATE 1,L:PRINT STRING$(19,143
):NEXT
50 FOR A=0 TO 3:X=10+A*180
60 PRINT CHR$(23);CHR$(A)
70 FOR P=0 TO 3:PLOT X+P*10,200,P
80 DRAWR 0,400
90 NEXT:NEXT
100 PRINT CHR$(23);CHR$(0)
110 PRINT TAB(6);"XOR";TAB(12);"AND";TAB
(18);"OR":PRINT:PEN 1
120 PRINT "0";TAB(7);"1";TAB(13);"2";TAB
(18);"3":PRINT
130 INPUT"ENTER ",R$
140 MODE 1
```

LISTING 3

```
10   CHR1 autres caracteres graphiques
20 MODE 1:PEN 2
30 PRINT TAB(6);"PRINT CHR$(1);CHR$(code
 ASCII)":PEN 1
40 T=-8
50 FOR N=0 TO 39 STEP 10:T=T+10:L=1
60 FOR J=0 TO 9:C=N+J:L=L+2
70 LOCATE T,L:PRINT USING"## ";C;
80 PRINT CHR$(1);CHR$(C);" ";
90 IF C<32 AND C>0 THEN PRINT CHR$(C+64)
 ELSE PRINT
100 NEXT:NEXT:PRINT
```

• **BOB MORANE A L'ECRAN :** *la présentation de Bob Morane n'est plus à faire , en effet, il a été largement vedette dans tous les domaines : romans, B.D. et feuilletons télévisés . Et maintenant, il se propose d'apparaître sur les écrans de vos micros grâce à une idée originale qu'Infogrames nous a développé lors de la présentation des produits , à cette occasion, nous avons eu la possibilité, rare il faut le noter, de réunir tous les programmeurs qui ont travaillé à ce projet...*
Le principe de la série Bob Morane est la suivante : présenter un coffret où le logiciel en lui-même ne constitue qu'une partie du jeu , en effet, en plus de la disquette, vous avez un roman de Bob Morane, une B.D. couleur, un guide de l'environnement où vous vous trouvez (jungle, espace ou chevalerie) et un jeu de rôle... Alléchant, n'est-ce-pas ?
Pour l'instant, les deux titres disponibles sur CPC sont les suivants : Bob Morane Science-Fiction où vous devez libérer votre ami Bill Ballantine qui se trouve sur une base de l'espace intersidéral. Au niveau du principe de jeu, il y a une certaine ressemblance avec Prohibition .
Le second titre étant Bob Morane Chevalerie, vous vous retrouvez dans un château du XIII• siècle où vous devez récupérer le "Saint Suaire" , mais, bien entendu, vous devrez pour cela affronter votre ennemi éternel qui n'est autre que l'Ombre Jaune...
La troisième histoire se déroulera dans la jungle et sera également un jeu d'arcade comme les deux précédents, ce qui est un peu dommage étant donné toutes les aventures que peut vivre un héros comme Bob Morane...



BOB MORANE





• **JOURNES ANTILLAISE :** le 28 novembre dernier, il suffisait de se rendre à l'auditorium de la FNAC du Forum des Halles pour respirer une petite bouffée d'exotisme !..
En effet, pour saluer la sortie de leur jeu d'aventure antillais MEWILO, Coktel Vision avait organisé un grand concours-spectacle récompensé de nombreux prix dont un voyage-séjour pour deux personnes à la Martinique. Tout au long de la journée, l'animation était assurée par le groupe Al Lirvat et des ballets antillais, le tout dans un décor de poupées et de costumes antillais.
Le logiciel en lui-même vous fait vivre une aventure dans l'atmosphère dramatique de St-Pierre de la Martinique menacée au début du siècle par les caprices du volcan. Pour l'instant il est disponible sur PC et compatibles au prix de 250 F, et ne va pas tarder à sortir sur CPC au prix de 220 F la disquette.

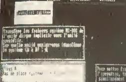


• **SIGNATURES DE CHOCS ! :** l'année 1988 commence très fort pour Gremlin Graphics et GO !..
En effet, figurez-vous que Mickey Mouse va souffler allègrement ses 50 bougies ! Pour l'occasion, il a décidé d'être présent sur les écrans de micros en signant un contrat de 2 ans avec Gremlins Graphics. Le premier jeu devrait sortir fin mars 1988 et il se déroulera dans le château de Walt Disney...
Quant à GO ! il a tout simplement signé des accords de licences pour 15 mois avec CAPCOM, le géant japonais des jeux d'arcade. Ces accords donnent à GO ! les droits de conversion et de distribution pour tous les types de micros sur 10 jeux d'arcade de Capcom dont 5 sont déjà connus ; il s'agit de Street Fighter, Black Tiger, 1943, Bionic Commando et Tiger Road. Le premier titre devrait sortir au printemps.

• **LYON CLUB :** le L.I.C. (Lyon Informatique Club) a pour but de démystifier l'informatique. Pour cela les débutants comme les experts auront à leur disposition une bibliothèque, une logothèque, une documenthèque, ainsi que des cours, des micro-ordinateurs en self-service. La cotisation annuelle est de 250 F. L.I.C. 138, cours Lafayette, 69003 LYON. Tél. 78.95.06.66.

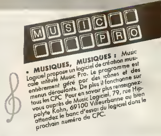
• **LOCOSCRIPT II :** l'utilitaire de traitement de textes réservé au PCW et livré avec le 9512 va être diffusé par la jeune et dynamique société Loisitech à Montreuil, suite à une exclusivité accordée par Locomotive System. Pour tous renseignements téléphonez au 48.59.72.76.

• **DOMICRO :** le MS-DOS vous rebute ? Vous ne comprenez pas toujours ce que vous tapez ? Menu est un utilitaire autochargeable qui enregistre une fois pour toutes les différentes commandes nécessaires au lancement de votre application. Tout est récapitulé dans un menu de 16 titres : il vous suffit de choisir parmi ces options. Didactidos se propose lui de vous apprendre, à l'aide de messages en français, les commandes de DOS. Chaque fois qu'il est nécessaire un appel à Didactidos vous donnera une série de commandes exécutables directement. Ces deux programmes tournent sous compatibles PC, XT et AT. Ils sont disponibles chez Domicro 16 (1) 48 42 52 10 au prix de 490 F H.T. l'unité.



• **UN BON CONSEIL :** le logiciel Conseil Plus est un programme qui gère les notes d'un ensemble d'élèves. Les données sont utilisées sur le nanoréseau et peuvent être lues, grâce à un programme spécial, sur un compatible PC. Les notes sont visualisées sous forme graphique ou imprimées sur papier. Ce produit est distribué par LOG'EDIC (84 76.04 01)

• **LES NOUVEAUTES DE L'APC :** l'APC ou Association de Presse et Communication nous communique les 3 nouveautés qu'elle propose pour ce début d'année 1988 : tout d'abord un guide gratuit de 96 pages d'informations pratiques sur l'Amstrad qui s'intitule "Le Guide de l'utilisateur d'Amstrad", vous pouvez vous le procurer par correspondance en joignant 3 timbres à 2,20 F. Ensuite, l'APC propose un logiciel serveur pour CPC 6128, AMSERV, comprenant une interface détection de sonnerie, un manuel d'exploitation et un logiciel permettant de piloter le minitel à partir du BASIC.
Enfin, sachez que l'APC organise dans ses locaux "la 1ère bourse aux logiciels" qui se déroulera du 20 au 23 janvier 1988. Il est à noter que l'entrée est gratuite et qu'un cadeau surprise sera remis à chaque visiteur !
Pour tout renseignement complémentaire, contactez :
A.P.C.
7, rue du Capitaine Ferber
75020 PARIS
Tél. (1) 48 97 84 84



• **10000 FRANCS DE PRIX :** attention, surveillez attentivement la sortie du CPC de février. Un concours y sera organisé par Petrel informatique. Des questions (faciles) vous seront posées et un tirage au sort désignera les heureux élus parmi les bonnes réponses. Le premier prix sera un Mav-save (un appareil magique qui conserve vos données pendant une coupure de courant) et les 9 suivant seront des logiciels utilitaires. Il est conseillé d'avoir sous la main quelques anciens numéros de CPC cela peut toujours servir. Tous les détails le mois prochain.

• **LORICIELS :** si vous avez un fort besoin de défoulement et que vous désirez vous affronter avec un jeu d'arcade dans l'espace se déroulant à une vitesse folle, montez dans votre vaisseau spatial et plongez avec **MACH 3 !**
Prix : K7, 140 F ; D7, 198 F



• **INFOGRAMES :** vous avez maintenant la possibilité de retrouver sur vos écrans Dilatlarath, le bon Haroun El Poussah et l'ignoble **IZNOGOUD** qui veut toujours être calife à la place du calife .. Ce logiciel d'arcade entièrement graphique est livré avec une B D d'Iznogoud (beau cadeau pour les étrennes !)
Prix : K7, 200 F ; D7, 250 F



• **ERE INFORMATIQUE :** que diriez-vous de vivre une nouvelle aventure de Crafton et Xunk ? C'est maintenant possible avec **L'ANGE DE CRISTAL** qui vous plonge dans l'arcade-aventure en 3D dont vous ne sortirez pas tout de suite...
Prix : K7, 140 F , D7, 220 F.



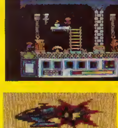
• **UBI SOFT :** depuis le temps que vous les attendez, vous avez cette fois deux logiciels à vous mettre sous la dent !... Il s'agit de **L'ANNEAU DE ZENGARA**, aventure où vous allez retrouver la signature de Fer et Flamme (les connaisseurs ont apprécié !) et de **PEUR SUR AMITYVILLE** où vous risquez de vivre un moment d'enfer dans votre nouvelle maison...







: entreprenez la visite d'un micro-monde artificiel et rentrez de plain-pied dans un jeu d'aventure graphique en 3D avec **INQUISITOR : Shade of Swords** où la progression dans l'aventure est assurée par une gestion d'icônes



• **DOMARK :** est une adaptation de jeu de café, une ! Avec **STAR WARS**, entrez dans le feu de l'action en plein espace...



• **CARRAZ EDITIONS :** du côté éducatif, les petits vont pouvoir faire connaissance avec **LES VOLEURS DU TEMPS** où Abracadabra, qui est un petit magicien chargé de la garde des éléments du temps, a quelques petits problèmes

• **GREMLIN GRAPHICS :** parmi la production toujours très prolifique de Gremlin Graphics, il est à noter la sortie de **MASK II** où vous retrouvez tous les courageux agents de l'équipe Mask ainsi que l'arrivée de la plus grande souris détective qui puisse exister. J'ai nommé : **BASIL THE GREAT MOUSE DETECTIVE !**

• **US GOLD :** Vous avez aimé Gauntlet ? Vous vous êtes éclaté avec The deeper dungeons ? Eh bien, vous allez pouvoir récidiver avec l'apparition de **GAUNTLET II** et retrouver Thor, Thyra, Merlin ou Questor.

• **PALACE SOFTWARE :** si vous acceptez de faire partie de l'équipe **STIFFLIP ET CO**, vous devez absolument empêcher le rayon caoutchoutronic de faire des ravages tels que dérider les flegmatiques britanniques !... Ça promet !





• **ELITE** : retrouvez les sensations du jeu de café dans l'épreuve de conduite la plus périlleuse qui soit avec **BUGGY BOY** (vitesse et tonneaux assurés...)



• **GO !** : avec **CAPTAIN AMERICA**, vous devenez l'Homme qui est le symbole de la liberté et du rêve américain et devez pénétrer dans le "Tube maudit" pour sauver les U.S.A. ; quant à **BRAVESTARR**, il nous vient encore des Etats-Unis car il s'agit d'un dessin animé qui devrait arriver bientôt sur les écrans (de TV cette fois !) pour remplacer Les Maîtres de l'univers

Enfin, pour terminer, il faut noter deux nouveaux titres chez **MASTERTRONIC** qui sont **SPOOK SCHOOL** et **AGENT X II** ; de même, **KONAMI** annonce deux autres produits : **IRON HORSE** et **JACKAL**.

Du côté des compilations, voici ce que les différents éditeurs proposent :

• **COKTEL VISION** : Carré d'As (Dakar 4×4, J. Debug dans le mystère de Paris, Starting Blocks, Balade au pays de Big Ben)

• **LORICIELS** : — Sélection Or (L'aigle d'or, Le 5e axe, MGT, Sapiens)
— Les hits de Loriciels 5 (Budget familial, Longraph, Space Shuttle)

• **HEWSON** : Four Smash (Zynops, Exalon, Rana Rama, Undium Plus)

• **TITUS** : Classiques n° 2 (Penggo, Arnold, Grand Prix)





BRANCHER LE TURBO

Yves GERAULT

4e PARTIE

LES TABLEAUX



Dans les précédents articles, nous avons développé les trois structures qui forment la base des langages informatiques : la séquentialité, la répétitivité, la conditionnalité. Pour la micro-informatique il n'en existe actuellement pas d'autres. En passant, nous avons introduit un outil propre à l'informatique : L'INVARIANT. Nous allons maintenant nous attacher à en développer d'autres. Ils n'ont tous qu'un but : RENDRE LA PROGRAMMATION PLUS FACILE ET LES TEXTES SOURCES PLUS CLAIRS EN OBLIGEANT LE PROGRAMMEUR A PLUS DE RIGUEUR DANS SON ANALYSE.

Certaines parties de programme peuvent être exécutées plusieurs fois de suite grâce aux instructions de répétitions. Pour cela nous avons introduit les INVARIANTS de structure, mais nous nous apercevons rapidement qu'il nous manque un outil pour vraiment pouvoir utiliser pleinement cette possibilité : les INVARIANTS de variables.

Reprenons notre problème simple d'addition de nombres. Nous l'avons résolu avec l'hypothèse qu'il n'était pas nécessaire de mémoriser ces derniers pour une utilisation ultérieure. En réalité parce que nous n'avions pas l'outil nécessaire pour le faire : LES VARIABLES INDICEES.

Essayons de solutionner notre problème, en gardant en mémoire les nombres entrés pour nous en servir plus tard. De plus, pour simplifier le problème, limitons-nous à 3 nombres ; il sera facile de généraliser à N nombres.

La structure de répétition que nous avons utilisée la dernière fois

```
for i.= 1 to 3 do readln (nombre);
```

ne peut répondre à notre attente puisqu'il n'y a pas mémorisation des nombres entrés, il nous faut faire :

```
readln (nombre 1,nombre 2,nombre 3),
```

Nous perdons le bénéfice de l'itération. De plus, cette technique est inutilisable dans le cas d'un grand nombre d'entrées. En regardant de très près notre ligne nous apercevons un INVARIANT qui pourrait s'écrire NOMBRE et une partie qui change et qui n'est qu'un numéro d'ordre **1, 2, 3.**

Au lieu de l'écrire NOMBRE 1, nous pourrions décider de l'écrire NOMBRE [1] et demander à l'ordinateur de réserver non plus UNE place mémoire de LONGUEUR définie par le type avec comme identificateur NOMBRE, mais PLUSIEURS places mémoires qui seraient atteintes par l'identificateur NOMBRE auquel il serait ajouté un décalage égal à I*LONGUEUR définie par le type de variable utilisée. Nous venons de créer les variables indicées.

Reprenons un exemple avec des chiffres afin de bien montrer que la partie de déclaration demandée par PASCAL pour créer ces genres de variables est une facilité pour le programmeur et non pas une contrainte inutile. Supposons que la variable NOMBRE a pour adresse en mémoire 8000 et que le type de variable utilisée nécessite 4 octets pour y loger l'objet manipulé. NOMBRE [0] sera à l'adresse 8000, NOMBRE [1] sera à l'adresse 8000+4=8004 et NOMBRE de 10 sera à l'adresse 8000+10*4=8040. Il est donc bien indispensable de préciser au compilateur qu'il doit réserver dans l'espace mémoire la place suffisante pour y loger plus tard toutes les composantes de la variable NOMBRE. C'est ce que fait d'ailleurs l'instruction DIM du BASIC. Mais PASCAL vous offre beaucoup plus, et pour cela il vous en demande un peu plus. Et si vous voulez adresser NOMBRE [-10] celui-ci devrait être en mémoire 8000-10*4=7960. PASCAL l'autorise mais comme il gère au mieux son espace mémoire, et qu'il ne peut pas prévoir si oui ou non vous allez utiliser cette possibilité, il va demander à ce que vous lui précisiez dans la partie déclaration les limites extrêmes que vous voulez pour le décalage que nous appellerons désormais au niveau PASCAL indice.



Tableau 1 - Syntaxes de l'instruction ARRAY.

PREMIERE MANIERE

```
type <ident_1> = array [ <indent_2> . <indent_3> ] of <type connu>;
ver <ident_4> : <ident_1> ;
```

DEUXIEME MANIERE

```
ver <ident_1> = array [ < indent_2> .. <indent_3> ] of <type connu>;
```



DECLARATION D'UN TABLEAU

PASCAL offre deux possibilités pour s'informer de vos désirs
Première possibilité : declaration directe d'une variable, ou d'une liste de variables ayant la structure TABLEAU
Deuxième possibilité : création d'un nouveau TYPE
Nous utiliserons cette dernière possibilité de préférence à la première pour les raisons suivantes
— cela permet de définir un nouvel outil très performant ,
— cela nous fera manipuler de nouveaux TYPES, apprentissage utile pour la suite ,
— dans un langage il convient d'utiliser toutes les possibilités de structuration que celui ci autorise Ce qui se conçoit bien s énonce clairement : une programmation confuse dans un texte source est un signe de mauvaise analyse du problème ,
— cela permet la modification rapide du dimensionnement des tableaux pour faire évoluer votre programme un programme est un produit qui évolue et qui necessite de fréquentes mises à jour
— Cela deviendra une nécessité lorsque nous nous servirons des PROCEDURES Alors autant commencer tout de suite et prendre de bonnes habitudes
L'opération, représentée au Tableau I, va se passer en deux temps
— **Dans un premier temps** nous créons un nouveau type.
— **Dans un deuxième temps** nous utilisons ce nouveau type
La création d'un nouveau TYPE se fait par la mise en œuvre d'un nouveau sous-bloc dans la partie déclarative Celui ci est introduit par le mot clef TYPE et, comme pour tous les sous-blocs de cette partie, il n'y a pas de mot clef pour le fermer , dommage ! Il est suivi d'un identificateur au sens PASCAL du terme, du signe EGAL et d'une partie descriptive Celle-ci commence par le mot clef ARRAY, qui signifie tout simplement tableau en anglais il est suivi par une expression comprise entre deux crochets (et non des parenthèses) Cette expression nous est déjà connue puisqu'il s'agit d'une désignation d'intervalle (rappelons toutefois que les deux points doivent impérativement être accolés) Ce descriptif se termine par le mot clef OF suivi d'un identificateur représentant un type déjà connu par le compilateur lorsque celui ci le rencontre lors de son travail de compilation (traduction du texte source ASCII en BINAIRE)
Ce nouveau TYPE s'utilise comme ceux que nous connaissons déjà C'est d'ailleurs une caractéristique du langage PASCAL IL EST EVOLUTIF, au gré des besoins de l'utilisateur Nous créons un nouveau type et il devient une primitive du langage De même, nous créerons de nouvelles procédures ou fonctions qui, elles aussi, deviendront des primitives du langage et ainsi de suite La seule limite est la taille mémoire de notre micro ordinateur, et encore !!! Mais n'anticipons pas Revenons un instant sur le signe EGAL qui se trouve entre l'identificateur et le descriptif dans le sous bloc TYPE Vous pesterez souvent contre ce signe qui n'est pas le < > habituel du sous-bloc VAR Il est pourtant logique car la proposition que vous venez d'écrire peut se lire de la manière suivante partout dans le texte source ou se trouvera l'identificateur ainsi nommé il faudra le remplacer à la compilation par le descriptif situé à droite Ceci est à rapprocher du même signe EGAL qui sépare l'identificateur de sa valeur dans le sous-bloc CONST, le fonctionnement en est le même
De ce fait la deuxième manière de déclarer un tableau devient évidente et est figurée dans le tableau I Elle présente l'avantage d'une apparente simplicité mais à l'usage elle rend le programme, dans le meilleur des cas, difficilement modifiable, et dans le cas de l'utilisation des fonctions et procédures inextricable

UTILISATION DES TABLEAUX

Cette utilisation découle directement du principe de cet outil Nous utiliserons chacun des éléments de ce tableau comme une variable Nous pourrons par exemple (listing 1) résoudre le problème que nous nous posions au début de cet article additionner et mémoriser N nombres entrés au clavier La particularité essentielle de ce programme réside dans le paramé modification de ceux ci dans le sous-bloc CONST réactualise tout le programme Il convient en outre de revenir sur le fait que les identificateurs attribues aux variables doivent être le plus explicites possible, cela complique la saisie du texte source mais facile la possibilite de mise à jour ultérieure mais facile la possibilité de mise a jour ultérieure
Le listing 2 nous montre ce qu'il est possible et impossible de faire avec les variables tableau A part le cas très particulier de la recopie intégrale d'un tableau dans un autre STRICTEMENT identique du point de vue TYPE les tableaux n'ont pas d'existence en tant qu'entité dans le PASCAL TURBO.On intervient toujours au niveau de l'élément de ce tableau, que ce soit pour le remplir, pour le lire ou pour permuter des valeurs à l'intérieur de lui même De ce fait il ne peut être question de faire du traitement de texte avec des variables de ce type Il faudrait écrire des procédures pour permettre une manipulation commode comme le permet le BASIC L'équipe BORLAND a eu exactement la même idée lors de l'écriture de ce compilateur et a mis au point un ensemble de procédures qui permet au PASCAL TURBO de pouvoir aisément rivaliser avec le BASIC à ce sujet Cependant pour des applications très pointues il sera quelque fois nécessaire de revenir à ce type de variable, en particulier lorsque le nombre de caractères sera teres sera trop grand , nous en reparlerons plus loin A titre d'exemple les deux derniers alinéas du listing 2 montrent comment il est possible de travailler sur ce genre de tableau de caractere pour faire du traitement de texte

LES CHAINES DE CARACTERES

Une chaîne de caractéres est une suite de caractères pris parmi les 256 possibles pour un micro B bits En PASCAL

TURBO ces chaînes forment un TYPE prédéfini. Comme l'annonce le manuel, ce type ressemble fort au TYPE ARRAY. De ce fait il en a presque tous les avantages, mais il en diffère aussi notablement car c'est un TYPE STRUCTURE, ce qui permet notamment d'avoir un nombre de caractères variant dynamiquement dans la chaîne. Puisque c'est un type il nous faut regarder de près les objets manipulables et les actions possibles sur ces objets.

OBJETS ET ACTIONS DU TYPE STRING

Le type chaîne (string en anglais) est un tableau de caractères particulier. Tout d'abord le minimum pour l'indice est 1, le maximum est laissé à l'appréciation de l'utilisateur et doit faire l'objet d'une déclaration. La position 0 est aussi utilisée, elle contient un caractère dont l'ordinal est la longueur de la chaîne. Les objets manipulés par le type STRING peuvent être des CONSTANTES ou des VARIABLES et sont assimilables à une juxtaposition de caractères. En conséquence, les actions possibles sont celles que nous avons définies pour le type CHAR.
— comparaison : l'ensemble est ordonné et correspond à l'ordre alphabétique, ce qui est commode pour les tris ;
— affectation : l'élément 0 représentant la longueur de la chaîne, s'ajuste automatiquement à la bonne valeur.
De nombreuses procédures et fonctions sont fournies par le logiciel pour manipuler commodément ces chaînes. Nous y reviendrons dans l'article prochain. Cependant nous retiendrons comme action élémentaire la concaténation puisqu'elle est représentée par la notation de l'addition, comme en langage BASIC. On peut d'ailleurs penser que c'est à l'image du BASIC que les concepteurs de PASCAL TURBO ont choisi ce symbolisme.

DECLARATION DE STRING

Le tableau II donne la syntaxe de cette opération. Il est à noter qu'il y a, là aussi, deux manières de déclarer ces chaînes, par analogie avec le type ARRAY. Comme pour lui, et pour les mêmes raisons, nous préconiserons la première méthode. Comme il n'y a que 256 caractères définissables sur huit bits, et que dans la première case (0) de notre STRING il y a un caractère dont le code ASCII donne la longueur de la chaîne, il est de ce fait évident que la longueur maximum de toute chaîne manipulée par TURBO PASCAL de type STRING est limitée à 256-1 (pour la case D) = 255 caractères. Et pour celles qui sont plus longues ? Il faudra soit les couper en morceaux, soit utiliser les tableaux et réécrire les procédures de traitement.

UTILISATION DES CHAINES

Le listing 3 donne un aperçu des possibilités de PASCAL TURBO en la matière en se bornant à la comparaison, l'affectation et la concaténation. La possibilité de considérer une chaîne de caractères à la fois comme un tableau, dont chacun des éléments est adressable, et comme une entité manipulable d'un bloc comme un chiffre, permet toutes les fantaisies en la matière. Il faut cependant se méfier de la case 0, si vous y mettez n'importe quoi, votre chaîne ressemblera aussi à n'importe quoi.

Tableau II - Syntaxes de l'instruction STRING.

PREMIERE MANIERE

```
type <ident_1> = string[ <ident_2> ];
var <ident_4> : <ident_1> ;
```

DEUXIEME MANIERE

```
var <ident_1> = string[ < INDENT 2> ];
```

Tableau III - Syntaxes de l'instruction ARRAY d'ordre N.

PREMIERE MANIERE

```
type
    str10    = string[10];
    tab1     = array [1 .. 5] of str10;
    tab2     = array [2 .. 8] of tab1
    tab3     = array [3 .. 7] of tab2
var
    tableau : tab3 ;
```

DEUXIEME MANIERE

```
type
    str10    = string [10];
    tab3     = array [3 .. 7, 2 .. 8, 1 .. 5] of str10
var
    tableau : tab3;
```

GENERALISATION DES TABLEAUX

Si nous nous reportons au Tableau 1 nous y voyons que la déclaration de TYPE se définit en fonction d'un TYPE déjà connu. De plus nous avons dit que chaque TYPE défini par l'utilisateur devient dès sa déclaration un TYPE à part entière pour PASCAL. Rien ne nous empêche donc de construire des type très complexes du genre tableau de tableau de ... de chaîne de caractères. Le tableau III donne un exemple de genre de tableaux imbriqués. Le compilateur du TURBO PASCAL se situera facilement dans votre structure grâce à sa notion de décalage mais cela est moins évident pour l'utilisateur. Afin de vous faciliter le travail, tableau III, PASCAL vous autorise à écrire cette déclaration MULTIPLE d'une autre manière qui permet une utilisation plus simple dans le programme. Le listing 4 vous donne un exemple d'utilisation d'un tel type de tableau d'ordre 2. Nous y voyons que les deux manières de déclarer les tableaux sont totalement équivalentes. Pour ceux qui connaissent les vecteurs, nous pouvons appeler un tableau à une dimension : un vecteur. Un tableau à deux dimensions serait un vecteur dont chaque élément serait un vecteur et ainsi de suite.
Pour clore ce chapitre, signalons une petite particularité toute à fait logique au niveau du compilateur, mais troublante pour l'utilisateur non averti. Si nous définissons une variable par la déclaration suivante :
VARIABLE : array [1 .. 5] of string [10]
Le 3e caractère du 4e élément du tableau VARIABLE doit être repéré par le symbolisme
VARIABLE [4][3] et non VARIABLE [4,3]
Cela semble en contradiction avec ce que nous avons écrit plus haut au sujet des tableaux du listing 4. En fait il n'en est rien, car, si les variables de type STRING ont un traitement analogue au tableau quant au niveau langage, il n'en est pas de même au niveau stockage en mémoire.



Listing 1 - Exemple de déclaration et d'utilisation d'un tableau.

```
program somme_nombre_et_memorisation (input,output);
{=====================================================}

const
     min_1   =   1;
     maxi_1  =   10;
{pour changer le tableau il suffit de modifier les valeurs 1 et 10}

type
     tab_int_maxi_1  = array [min_1 .. maxi_1] of integer;

var
     i,somme        :  integer;
     nombre         :  tab_int_maxi_1;

begin
  somme:=0;
  for i:=min_1 to maxi_1 do
    begin
      write('entrer le nombre numero ',i:4,' : ');readln (nombre[i]);
      somme:=somme+nombre[i];
    end;
  for i:=min_1 to maxi_1 do write (nombre[i]:4);writeln;
  writeln ('la somme est : ',somme:5);
end.
```

Listing 2 - Exemple illustrant les diverses actions possibles sur les tableaux.

```
program manipulation_tableau (input,output);
{==========================================}

const
        min = -5; max = 5; {extremun des tableaux}

type
        tab_char_min_max = array [min .. max] of char;

var
        car             : char  ;
        tabin,tabout    : tab_char_min_max;
        i               : integer;

begin
{on initialise les deux tableaux avec des blancs}
{pour  remplir tabin et tabout on doit agir element par element }
  for i:= min to max do
    begin tabin [i]:=' ';tabout[i]:=' ';end;

{on remplit le tableau tabin au clavier}
  for i:= min to max do
    begin write ('entrer caractere : ');readln (tabin[i]);end;

{on ecrit sur l'ecran le contenu des tableaux tabin et tabout   }
{on ne peut ecrire d'un bloc tabin ou tabout sur l'ecran        }
{writeln (tabin);writeln (tabout); est illegal                  }
{on est oblige d ecrire element par element. C'est un peu lourd }
  for i:= min to max do write (tabin [i]);writeln;
  for i:= min to max do write (tabout[i]);writeln;

{on remplit tabout a partir de tabin caractere par caractere}
  for i:= min to max do tabout[max+min-i]:=tabin[i];

{on ecrit sur l'ecran le contenu des tableaux tabin et tabout   }
  for i:= min to max do write (tabin [i]);writeln;
  for i:= min to max do write (tabout[i]);writeln;

{on remplit le tableau tabout d'un bloc a partir de tabin}
  tabout:=tabin;

{on ecrit sur l'ecran le contenu des tableaux tabin et tabout   }
  for i:= min to max do write (tabin [i]);writeln;
  for i:= min to max do write (tabout[i]);writeln;

{on ecrit BON dans tabin et JOUR dans tabout et concatene dans tabout}
  for i:=min to max do  tabin[i]:=   ';tabout:=tabin;
  tabin [min]:= B';tabin [min+1]:='O ;tabin [min+2]:='N';
  tabout[min]:='J';tabout[min+1]:='O';tabout[min+2]:='U';tabout[min+3]:='R';

   {tabout:=tabin+tabout; est illegal. On doit agir element par element}
    for i:=min+3 downto min  do tabout[i+3]:=tabout[i];
    for i:=min to min+2      do tabout[i]:=tabin[i];
    for i:=min to max do write (tabout[i]);writeln

end.
```

Listing 3 - Exemple illustrant les diverses actions possibles sur les chaînes de caractères.

```
program chaine (input,output);
{==============================}

type
       str30    =  string [30]; {declaration du maximun 30}

var
       nomin,nomout          :    str30;
       i,j,longueur          :    integer;

begin
{on rentre nomin au clavier et on l'affecte a nomout}
  write ( entrer un nom :  ');readln (nomin);
  nomout:=nomin;

{on ecrit sur l'ecran nomin et nomout}
  writeln(nomin:35,nomout:35);

{on affecte chaine nulle a nomout}
  nomout:='';

{on determine la longueur de nomin }
  longueur:= ord (nomin [0]);

{on remplit nomout caractere par caractere}
  for i:=0 to longueur do nomout[i]:=nomin[i]; {ne pas oublier le 0}
  writeln (nomin:35,nomout:35);

{on remplit nomout par concatenation successive}
  nomout:='';
  for i:=1 to longueur do nomout:=nomout+nomin[i];{ne pas commencer a 0}
  writeln (nomin:35,nomout:35);

{on concatene nomin et nomout et on affecte a nomout}
  nomout:=nomin+nomout;
  writeln (nomin:35,nomout:35);

{on effectue une troncature a 5 caracteres}
  nomout[0]:=chr(5);writeln (nomin:35,nomout:35);

end.
```

Listing 4 - Exemple de déclaration et d'utilisation des tableaux multiples.

```
program tableau_ordre_2 (input,output);
{======================================}

const
      max1 = 3; min1 = 1;
      max2 = 9; min2 = 0;
type
      tab_min1_max1            = array [min1..max1] of integer;
      tab_min2_max2            = array [min2..max2] of tab_min1_max1;
      tab_min2_max2_min1_max1  = array [min2..max2,min1..max1] of integer;
var
      tab1     : tab_min2_max2;
      tab2     : tab_min2_max2_min1_max1;
      i,j      : integer;

begin
  {remplissage des elements d'un tableau}
  for j:=min2 to max2 do for i:=min1 to max1 do tab1 [i,j]:=i*10+j;

  {egalite de deux tableaux}
  for j:=min2 to max2 do for i:=min1 to max1 do tab2 [i,j]:=tab1 [i,j];

  {manipulation des elements des tableaux}
  for j:=min2 to max2 do for i:=min1 to max1 do
      tab2 [i,j]:=tab2 [i,j] div 10 + 10*(tab2 [i,j] mod 10) ;

  {impression des elements des tableaux}
  writeln;writeln(   tableau  1        tableau 2  ');writeln;
  for j:=min2 to max2 do
    begin
      for i:=min1 to max1 do write (tab1 [i,j]:4);
      write ('    ');
      for i:=min1 to max1 do write (tab2 [i,j]:4);
      writeln;
    end;
end.
```

COURRIER DES LECTEURS

POINT D'ENTREE

J. Thoris d'Orsoy nous demande comment il se peut qu'une routine en langage machine implantée de &9800 à &9fff, puisse être appelée par un CALL &9b00.

Vous aurez remarqué que nous avons employé un "b" pour l'adresse du CALL. Un "B" aurait ressemblé étrangement au 8 de l'adresse de début d'implantation. Supposez maintenant que vous venez de saisir un programme utilisant ces adresses. Le programme ne tourne pas et, vous vous dites aussitôt que l'adresse du CALL, &9B00 est fausse. A moins que ce ne soit l'autre, &9800, une erreur d'impression ayant fait confondre "8" et "B" ! Vous êtes sûr de votre fait : "Comment un programme peut-il être lancé à une adresse autre que celle de début ?". Un programme écrit en Basic démarre généralement à la première ligne d'implantation. Nous disons "généralement" car il est tout à fait possible de déplacer l'adresse d'entrée. En revanche, de nombreux programmes écrits en langage d'assemblage ont un point d'entrée différent de l'adresse d'implantation. Voyez d'ailleurs, la syntaxe de la commande de sauvegarde SAVE pour les programmes binaires.

SAVE "nomprog. ext", B, adresse d'implantation, longueur du programme, adresse du point d'entrée. Il va sans dire que lorsque "adresse d'implantation" et "point d'entrée" sont confondus, on peut omettre ce dernier.

PUISSANCE

Frédéric Sannier de St Orens est intrigué par le signe "^" qu'il retrouve régulièrement dans les programmes de CPC.

Ce signe est le signe d'élévation à la puissance d'un nombre. Ainsi, 2^3 donne pour résultat 8. L'inconvénient de ce signe est qu'il n'apparaît pas à l'écran, ni sur les listes de programmes, sous la forme qu'il prend au clavier où il est représenté par une flèche verticale.

SAISIE AISEE

Philippe Fouquerel d'Hérouville St Clair est las de saisir les programmes de son clavier et se demande s'il n'existerait pas des moyens moins contraignants.

On peut, en effet, imaginer des systèmes de saisie mettant en œuvre la synthèse vocale, il suffirait alors de lire le programme. Mais ce serait encore trop long. Il existe bien le SOFTSTRIP qui semble répondre parfaitement à la demande de Philippe. C'est un système équivalent d'un analyseur (scanner). Le programme apparaît sous la forme de bandes de quelques centimètres de large sur la hauteur d'une page de revue, où les données apparaissent sous la forme d'une sorte de codes-barres. Le SOFTSTRIP, autotracté, passe sur ces bandes de 1 Ko, les charge dans le micro qui les traite grâce à un logiciel implanté dans le micro. Le programme se trouve alors chargé. Il est également possible de sortir les programmes sur imprimante. Génial, non ? Mais c'est encore un peu lent. Il faut bien convenir que le meilleur moyen, le plus rapide, le plus sûr, est de se procurer les programmes sur support disquette !



ERREURS DANS L'ANTI-ERREURS

De nombreux lecteurs nous écrivent au sujet de l'anti-erreurs de CPC n° 27. Récapitulons. Version 1.0 jusqu'au n° 27, reconnaissable aux doubles crochets des codes de contrôle. Cette version est remplacée par une version plus performante dans le n° 27, la version 2.0. Malheureusement, une erreur s'y est glissée et elle est inutilisable. Une version corrigée est alors publiée dans le n° 28. C'est la version 2.1, elle génère un seul crochet pour les codes de contrôle comme la version précédente. Ne vous trompez donc pas !

TROIS DES

Pierre Ridoux nous avait écrit pour nous signaler que le cinquième tableau de TROIS DES ne fonctionnait pas. Il vient de nous réécrire pour nous signaler son erreur.

Que les lecteurs sachent que pour réussir le 5ème tableau, il ne faut pas être trop bon ! Nous ne vendrons pas la mèche mais, lorsque vous aurez trouvé la solution il faudra vous accrocher à votre siège !

UTILITAIRE

DEUX LOGICIELS DE

La P.A.O. on vous en rebat les oreilles. Partout dans les salons, les revues spécialisées on vous vante les mérites de ces merveilleux logiciels capables de réveiller le Gutemberg qui dort en vous. Mais peut-être que certains d'entre vous ne connaissent pas encore ce sigle et que P.A.O. n'a pas vraiment de sens. Voici donc quelques explications concernant la Publication Assistée par Ordinateur (voilà, maintenant vous connaissez la signification du sigle). Les logiciels de P.A.O. ont pour but la confection de bulletins, de notes ou même de petites revues de clubs. Je me limite ici uniquement aux programmes que l'on peut trouver sur le CPC. En effet si l'on considère des machines plus puissantes il est possible de réaliser de véritables revues. La meilleure preuve : la revue que vous tenez entre les mains est partiellement "fabriquée" à l'aide d'un Macintosh et d'une imprimante laser. Mais si les performances sont assez disparates (les besoins professionnels et amateurs n'étant pas les mêmes) l'interface imprimante laser n'étant pas prête de voir le jour sur CPC, en revanche l'"esprit" des différents logiciels est le même. Pour composer une page il faut : du texte et parfois du graphisme. Pour le premier il suffit d'un traitement de textes pour le second un logiciel de dessin ou un digitaliseur est le bienvenu. Le programme de P.A.O. n'est qu'un outil qui permet d'assembler et de manipuler les textes et les illustrations. Le principal problème posé par ce type de programme est la mémoire utilisée. En effet les graphiques principalement sont gourmands en mémoire, donc le 6128 est quasiment obligatoire (ou à la rigueur un 464 + extension mémoire) ainsi que le lecteur de disque et une imprimante en état de marche. Pour la facilité d'utilisation une souris est recommandée (AMX Mouse ou Kempston au choix). Le joystick est tout juste passable, quant aux touches du curseur...

Copyright: MEIROSOFT (C)

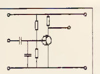
use

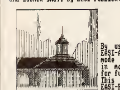
EASI PUBLISH

in combination with:

EASI-DRAW

By using the included tool to convert an EASI-DRAW drawing into a screendump, suitable for further manipulation in the PUBLISHER.
This circuit diagram has been converted and zoomed small by EASI-PUBLISH.





EASI-ART

By using the included tool to convert an EASI-ART drawing, which is designed in mode 1 with 4 colours, into a screendump in mode 2 with 'shaded' colours, suitable for further manipulation in the PUBLISHER.
This church has been 'cut and pasted' by EASI-PUBLISH.

Vous avez maintenant tous les éléments en main pour débuter votre œuvre. Un dernier point cependant, les deux logiciels décrits ci-dessous sont de provenance britannique : ils ne tiennent donc aucun compte des accents de notre chère langue. Il est tout de même préférable d'attendre la version française pour faire son choix.
Les deux logiciels sont Stop Press (d'AMS Software) et Easi Publish (de Métrotec).
Tout d'abord la présentation : Easi publish apparaît sous forme d'un classeur bleu contenant le logiciel et une dizaine de feuillets constituant la notice. Stop Press (désormais je l'appellerai S.P.) est conditionné dans une boîte de carton dans laquelle on trouve une disquette et un manuel à la reliure spiralée.
E.P. (pour Easi publish) est à utiliser conjointement (mais pas obligatoirement) avec Easi draw ou Easi art conçus tous deux par la même société. Easi Draw est un utilitaire de dessin technique et Easi art un logiciel de dessin artistique. A ce propos puisque Easi art fonctionne en mode 1 et E.P. en mode 2 vous trouverez un module de conversion qui traduit les 4 couleurs de la moyenne résolution en dégradés de vert pour la haute résolution.
Le menu principal vous offre 4 options : quitter qui se passe de commentaires, créer une page ou un jeu de caractères, convertir un dessin en mode E.P. et enfin effectuer le passage de la révolution 1 à la révolution 2.

L'option principale est bien sûr la création d'une page. Un appui sur la touche "1" et l'on vous demande le type d'outil employé (souris, trackerball,

P.A.O. SUR AMSTRAD

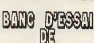
joystick, touches du curseur). A propos le trackerball est une sorte de souris posée à l'envers et qui se manœuvre à l'aide d'une grosse sphère.
Le deuxième sous-menu donne accès à l'éditeur de polices. Une police est un ensemble de caractères ayant les mêmes caractéristiques morphologiques. (Exemple : les lettres en italique)
L'accès à ce module n'est pas des plus pratiques puisque vous êtes obligé d'entrer un nom de fichier pour commencer l'édition (ou plutôt la correction). En effet il existe 7 séries de caractères différentes. Si vous avez l'ambition de créer votre propre jeu (bon courage !) il faudra donner un nom "bidon" pour avoir accès à une grille vierge. La taille maximum des caractères est de 6×3.
Revenons au module principal qui se compose d'une feuille "blanche" à côté de laquelle est inscrit un bandeau d'options.
Ces options concernent le format des lettres (standard ou en provenance d'un fichier ASCII).

Il y a également deux recopies d'écran : l'une au format A4 et l'autre occupant la moitié d'une feuille A4. La recopie est correcte mais il est préférable de disposer d'un ruban neuf pour obtenir un bon résultat.
Les options LOAD et SAVE concernant les spasges déjà constituées, les écrans ou les fenêtres. En effet il faut utiliser une fenêtre qui servira alternativement aux textes et aux graphismes.
Le principe de fonctionnement est très simple : sur l'un des deux écrans (le format A4 occupe 2 parties d'écran accessibles par la fonction scroll) on place une fenêtre ou l'on se sert directement de la page. Les textes peuvent être justifiés à droite ou centrés. Les graphiques sont constitués soit d'un écran créé auparavant, soit dessinés à la main grâce aux outils : boîtes, lignes, points et inversion vidéo. L'intérêt de la fenêtre est de permettre des copies des zooms (assez limités : il n'y a que deux tailles possibles), des rotations à droite ou à gauche. Il est vrai que le nombre d'outils proposés n'est pas faramineux et que les placements de colonnes de textes ne sont pas facilités par les deux écrans. De plus "la grille" proposée pour le repérage sur la page n'est pas modifiable : on est obligé d'utiliser le format du programme. E.P. est donc un programme simple (peut être même trop) dont l'utilisation se limite à la création d'une page agrémentée de graphismes. Le texte ne peut être entré qu'à la main directement sur la page. En cas d'erreur il faut tout effacer !
Pour plus de renseignements contactez directement Métrotec (vous trouverez l'adresse de la société dans le carnet de CPC).
Stop Press a lui, d'autres ambitions. Premier indice le programme et ses fichiers occupent les 2 faces d'une dis ▷



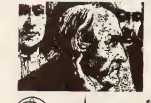
CCP

la revue de l'ASTRAM

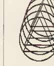


BANC D'ESSAI DE

STOP PRESS











UTILITAIRE

quette. Le nombre de polices disponibles est de 17. Chaque page enregistrée occupe 74 Ko ! (Un rapide calcul nous montre qu'une disquette ne peut contenir que 4 pages). Il existe une bibliothèque de dessins sur la disquette principale, cette bibliothèque pouvant être complétée par deux disquettes : l'une étant bourrée à craquer de dessins en tout genre et l'autre complétant les "fontes" de caractères grâce à 23 polices supplémentaires. Au vu de ces chiffres on s'aperçoit que S.P. est un véritable "monstre". Les options disponibles sont en nombre considérable et il me faudrait plus d'un CPC pour en venir à bout.

Un écran blanc entouré d'un code gradué en inches et en centimètres (s'il vous plaît) le tout surmonté d'un bandeau d'icônes, tel est le décor qui apparaît lors de l'utilisation de S.P. Le curseur est matérialisé par une petite flèche (bien sûr). Celle-ci se transforme en repère lorsqu'elle arrive sur la page blanche. L'écran ne représente qu'une fenêtre sur la page en taille réelle. Il existe heureusement une option permettant de visualiser (en taille réduite) la page entière. Deux types de fenêtres peuvent être ouverts : texte ou graphique. Mais attention un seul type de fenêtre peut être ouvert à la fois. Ceci posera des problèmes lors de l'entrée d'un texte. En effet sur les 3 boutons disponibles de la souris, le premier sert à sélectionner une option, le second délimite le premier coin des fenêtres, le troisième annule l'option choisie. Lors de l'entrée d'un texte par l'intermédiaire d'un fichier ASCII (issu de votre traitement de texte favori) le placement de ce texte se fait automatiquement dans la fenêtre. Mais si le texte est très long un message d'erreur apparaît vous demandant si la suite peut être inscrite dans une autre fenêtre. Il semble facile d'ouvrir un espace et de laisser "couler" l'écriture, en revanche si vous vous trouvez en bas de page il va vous falloir "remonter". Malheur, si vous cliquez sur la touche 2 (qui sert également au déplacement) vous validez le premier coin de la fenêtre de texte. Malgré ce défaut gênant il existe des fonctions intéressantes, telles les différentes justifications, l'habillage automatique, le nombre de polices disponibles. Du côté graphiques on dispose d'outils tels que tracé de cercles, de lignes, de points, un aérographe. Vous pouvez y ajouter la possibilité d'agrandir, de réduire, de copier, de scroller une image, de charger un dessin venant de n'importe quelle source pourvu que celui-ci soit en mode 0 et fasse 17 Ko. Plus tous les dessins présents dans les bibliothèques. Plus une option scanner (celui d'AMS) qui permet de traiter directement les images provenant d'une caméra ou une interface. Les sorties sur imprimantes sont particulièrement soignées avec les possibilités suivantes : recopie d'une page, d'un écran ou d'une fenêtre et ceci en trois densités différentes !

Puisque les images parlent mieux que les textes, les recopies d'écran vous donneront une petite idée des possibilités.

En conclusion, si la P.A.O. sur CPC n'est pas à utiliser dans une optique professionnelle en revanche un utilisateur amateur pourra fabriquer assez rapidement de petits documents de 1 à 4 pages avec une qualité correcte.

GRAPHIQUES +

Dans CPC n° 26 vous avez été séduit par le programme GRAPHIQUES de Thierry Bas, et vous l'avez saisi immédiatement sur vos claviers. Il va falloir vous y remettre pour ajouter cette amélioration de Yves Oury de Reims qui vous permet d'afficher sur vos graphiques les valeurs négatives. La ligne 1030 modifiée a pour but d'améliorer l'affichage pour les valeurs proches de 0, la ligne 1070 est là pour faire plus joli !
Pour obtenir des barres d'histogrammes alternativement positives ou négatives, ajoutez ou modifiez

```
 745 ZZ=MAX(ABS(ZX),ABS(ZY))
 750 ZZ$=STR$(ZZ).LZZ=LEN(ZZ$):LY=LZZ-4
 885 PLOT 40,190.DRAW 638,190
1030 NDON(I)=INT(DON(I)*160/ZZ)+190
1040 PLOT ES,190·DRAW ES,NDON(I).
     DRAW ES+24,NDON(I):DRAW ES+24,190
1045 IF DON(I)<0 THEN 1061
1050 PLOT ES+24,190.DRAW ES+32,198
     DRAW ES+32,NDON(I)+8:DRAW ES+8,
     NDON(I)+8.DRAW ES,NDON(I)
1060 PLOT ES+32,NDON(I)+8:DRAWES+24,
     NDON(I).GOTO 1070
1061 PLOT ES+24,190 DRAW ES+32,182·
     DRAW ES+32,NDON(I)-8 DRAW ES+8,
     NDON(I)-8·DRAW ES,NDON(I)
1063 PLOT ES+32,NDON(I)-8 DRAW ES+24,
     NDON(I)
1070 FOR COUL=0 TO 24 STEP 4
1080 PLOT ES+COUL,190.DRAW ES+COUL,
     NDON(I)
1200 MOVE -10,30.PRINT INT(ZZ/LYD);
1210 MOVE -10,360 PRINT INT(ZZ/LYD),
1220 MOVE -10,200:PRINT INT(0);
1230 MOVE -10,280 PRINT INT((ZZ/2)/LYD),
1240 MOVE -10,120:PRINT INT((ZZ/2)/LYD),
```

Du beau travail, Yves Oury de Reims ! Nous avons sous les yeux un histogramme des températures dans la région de Reims au cours de l'année 1986, c'est très parlant et esthétique

BIORYTHMES 464

Les quatrecentsoixantequatristes vont gagner un programme sur CPC HS n° 6 BIORYTHMES, qu'ils ne pouvaient utiliser, va, enfin, pouvoir tourner sur leur machine préférée. Ils n'auront rien, ou presque rien, à envier aux sixcentsoixantequatristes ni aux simillecentvingthuitistes. Presque rien, ce n'est pas rien. Ils ne pourront pas utiliser la sortie sur imprimante. Reste un petit coup de "Polaroïd" !
Le truc de Robert Coste de la Seyne sur Mer est simple comme "bonjour" ! Ils supprimeront toutes les commandes MASK, inconnues au bataillon, ainsi que les lignes : 1720, 1780 et 1910 à 2130.

A VOT' BON CŒUR !

Un lecteur aurait bien besoin de savoir s'il existe un compilateur COBOL tournant sur CPC. Saurez-vous le renseigner ?
Un autre lecteur s'attendait à trouver 32 caractères d'extension sur le clavier de son CPC 464 entre les valeurs 128 et 140. Il trouve bien sur le pavé numérique les valeurs 128 à 140, mais quid des touches 141 à 159 ? Répondez-lui, nous vous ferons l'honneur de ces colonnes et tout le monde en profitera !

TYPE MISMATCH

Les lecteurs qui saisissent les programmes de CPC aux DATAS abondants sont familiers de TYPE MISMATCH ERROR. Ils ne comprennent pas toujours ce qui s'est passé et recherchent indéfiniment dans les lignes de DATAS l'erreur qu'ils ont pu commettre
Voici un petit exemple de bogue qui génère ce message, proposé par Maurice Thion de Tarascon. Essayez de découvrir la solution avant de la lire dans les lignes qui suivent.

```
10 WHILE A!<>255
20 READ A$,A!
30 IF A!=73 THEN READ B$
40 PRINT CHR$(A!);
50 WEND
60 DATA A,65,B,66,C,67,D,68,E,69,F,70,G,71,
   H,72,I,73
70 DATA J,74,K,75,L,76,M,77,FIN,255
80 END
```

Voilà ! Vous avez saisi le programme, l'avez fait tourner et obtenu un beau message d'erreur en or massif ! Vous avez joué le jeu, et vous avez trouvé l'erreur, bravo, il ne faut pas vous la refaire !
Les débutants, et les cancres, en revanche, ont besoin de comprendre. Ils auront lu et relu les lignes de DATAS, et, malgré toutes leurs recherches, seront restés le bec dans l'eau. L'erreur est ailleurs, en ligne 30. Analysons le fonctionnement du programme. Son but est d'afficher la lettre de l'alphabet correspondant à la valeur des DATAS numériques. Le programme lit en 20 les DATAS deux par deux. Le premier est alphanumérique (A$), le second numérique (A!). Mais, en ligne 30, lorsque la valeur de A! est 73, il est demandé à la machine de lire B$. Que fait la machine? Elle lit la DATA suivante et la place dans B$, cette donnée alphanumérique. Il lit le caractère J. Voici ce qui se passe alors. En ligne 40, affichage du caractère ASCII 73 (I), retour en 10 pour comparer 73 et 255 (fin du programme ? Non). Lecture en 20 des deux données suivantes, et là, CRAC ! Ça ne marche plus ! Le pointeur de DATAS (petit programme de votre CPC qui est chargé d'enregistrer le dernier DATA employé) pointe le J. La prochaine donnée à lire est donc 74, donnée NUMERIQUE, or, le programme cherche à lire une donnée ALPHANUMERIQUE ! Aussitôt le système de votre CPC affiche le message d'erreur approprié : TYPE MISMATCH ERROR IN 70. Alors que la ligne 70 ne comprend pas la moindre erreur. Voilà qui vous éclairera dans la recherche des erreurs.

CATALOGUE AMELIORE

Vous utilisez certainement le programme d'impression de catalogues oublié dans CPC n° 27 que nous avait envoyé Maurice Tollu de Besançon. Voici une amélioration que lui apporte son auteur.
Remplacez les lignes 120, 130 et 140 par les suivantes :

```
120 PRINT #8,CHR$(27);"1",CHR$(10).
    IF A$="1" THEN IPRTCAT,0
130 PRINT #8,CHR$(27),"1",CHR$(10):
    IF A$="2" THEN IPRTCAT,1
140 PRINT #8,CHR$(27);"1",CHR$(3)
    IF A$="3" THEN IPRCAT,2
```

N'oubliez pas que la RSX appelée (IPRTCAT,X) figure dans CPC n° 14. Cette version facilitera l'impression sur papier perforé.

LOGICIELS

QIN
ERE INFORMATIQUE
Aventure

Nous sommes en 221 av. J.-C., période à laquelle l'empereur de Qin fait régner tout son despotisme. Il a chargé son plus grand architecte, Liou, de la construction de son tombeau et, suivant la tradition, celui-ci sera enterré avec l'empereur...

Seulement, avant d'être sacrifié, Liou a éparpillé le plan de la sépulture dans toute la Chine.
Et c'est ainsi que vous en tant que descendant de Liou, vous décidez 1000 ans plus tard de percer le secret de Qin ; tout commence avec un vase qu'il vous faudra casser et qui vous entraînera dans un long voyage allant du pic de l'Est jusqu'aux limites de la Grande Muraille de Chine. En comptant sur



les objets que vous récolterez et sur les rencontres que vous ferez, vous pourrez rassembler 3 morceaux du plan.
Seulement, il en reste encore 2 que vous ne pourrez récupérer qu'en étant transféré à l'époque de votre ancêtre soit 1000 ans plus tôt !.. (Heureusement que les dieux sont là pour vous donner un petit coup de pouce !). Alors, une fois le plan reconstitué, il vous faudra affronter les nombreux dangers se trouvant dans le tumulus avant d'espérer atteindre la chambre du sommeil de l'empereur et avoir ainsi la révélation totale du secret.
Réalisé par les auteurs de Sram, ce logiciel présente un graphisme toujours aussi agréable, un analyseur syntaxique encore plus souple et un scénario "solide" (2 disquettes).. alors pourquoi vous en priver ?



RAMPARTS
GO !
Arcade

Preux chevaliers en brillantes armures, vous qui défendez si courageusement les opprimés, la veuve et l'orphelin, vous dont la gloire.. Stop, vous n'y êtes pas du tout. Ramparts n'a rien d'une ode à la chevalerie et à ses principes. Ce serait même plutôt le contraire. Imaginez la douce campagne anglaise avec ses petits vallons verdoyants et ses châteaux miroitants (cela ne veut rien dire mais au moins ça rime). Ventrebleu deux gigantesques personnages viennent d'apparaître. Malheur ce sont les victimes de la malédiction du grand sorcier ; auparavant bien vaillants, ils sont devenus des pillards et des destructeurs. Voyez donc leur stratégie : ils grimpent le long des tours du château et ébranlent d'un coup de poing le bâtiment. A chaque





coup une portion du mur s'écroule laissant parfois apparaître un gâteau ou une bourse que ces deux gredins s'approprient immédiatement. Que pouvons-nous faire contre ces monstres ? Il ne nous reste que les armes conventionnelles : les flèches et les pierres. Malheureusement ces méthodes ne sont pas particulièrement efficaces mais à la longue les géants faiblissent et meurent (à condition qu'ils ne se nourrissent pas entre-temps). Contrairement à ce que vous pourriez penser, vous n'êtes pas dans la peau d'un des défenseurs mais dans l'armure du démolisseur. Ce qui est nettement plus réjouissant. Il s'agit presque d'une nouvelle version de casse-briques (si l'on peut dire) mais possédant des décors et une animation superbes. Go ! nous avait habitués à un graphisme soigné, il ne nous déçoit pas avec Ramparts qui allie originalité et esthétisme.

TURLOGH LE RODEUR
COBRA SOFT
Aventure

Tout d'abord, je commence par faire les présentations : je suis Turlogh, rôdeur mais néanmoins, cependant et par conséquent baron de Perroth. Le destin m'a fait rencontrer un jour au détour d'un chemin non pas une belle damoiselle mais le roi Ydahaut Ar Dinlach.



Après lui avoir été d'un grand secours dans la sphère du Nécromant (je le dis en toute modestie !), il fait de nouveau appel à mes services pour venir en aide à la cité delle de Much'Duaryat qui semble en très mauvaise posture...

C'est donc sans plus attendre que je prends la route avec pour tout équipement mon cheval, ma dague et ma fidèle épée. Je prends malgré tout le temps de faire un petit détour pour aller rendre une petite visite chez Achnea, magicienne de bon conseil. Lorsque je me trouve enfin devant la citadelle (ou ce qu'il en reste !), les combats commencent et ce n'est pas le nombre d'adversaires différents qui manque suivant l'importance de ma progression : les archeopterix, le dragon, les kobolds ou l'ours... (et encore, je ne dis pas tout !).





Avec Turlogh le rôdeur, vous avez malgré tout deux avantages : une superbe B.D. interactive vous permettant de vivre une belle aventure et le logiciel où vous retrouvez les mêmes héros dans 2 fenêtres présentant un graphisme correct mais sans animation ; la gestion en elle-même se fait souplement par icônes... Somme toute, un soft intéressant.

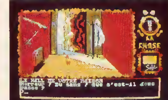


LA CHOSE DE GROTEMBURG
UBI SOFT
Aventure

Si vous voulez vivre une aventure avec un grand "A" et frissons dans le dos, il vous suffit de suivre les traces (d'hémoglobine, bien sûr !) qui se présentent sur votre écran.

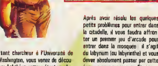
Ainsi, vous allez découvrir dans votre cuisine Germaine (ou plutôt ce qu'il en reste...) après le passage de la Chose ; le spectacle sera à peu près similaire dans la maison de vos voisins...

Si vous orientez correctement vos recherches, vous ne tarderez pas à rencontrer le dénommé Sergio et son message : "Dans la forêt tu pénètreras, le souterrain tu découvriras, au manoir tu accéderas, le savant fou tu rencontreras, mais méfie-toi des rats"... Vous voilà ainsi prévenu et informé ; conclusion de tout ceci : y a plus qu'à !... Cette aventure va vous en faire voir de toutes les couleurs, à commencer par le rouge, bien sûr ! Ensuite, vous allez évoluer dans des endroits aussi différents que les rues de la ville ou la forêt en passant par la décharge du coin !... Grâce à une fenêtre qui occupe les 2/3 de l'écran, vous pouvez apprécier un bon graphisme qui est secondé par un analyseur syntaxique suffisamment souple. Par contre, vous serez sûrement plus d'un à trouver l'humour un peu "lourd"...



LOGICIELS





Etant chercheur à l'Université de Washington, vous venez de découvrir Ackthéon comme étant un pharaon de l'antimatière, celui ci aurait trouvé dans son existence passée LA formule permettant de dissimuler certaines parties de la terre lors d'un éventuel conflit.

C'est donc sans plus attendre que vous vous rendez au Caire avec la ferme intention de trouver la formule, seulement, il ne va pas falloir oublier un seul instant que votre homologue bulgare Yvan Skivol a les mêmes objectifs que vous et qu'il n'hésitera pas un seul instant avant de vous faire "coffrer" !

Après avoir résolu les quelques petits problèmes pour entrer dans la citadelle, il vous faudra affronter un premier jeu d'arcade pour entrer dans la mosquée : il s'agit du labynum (ou labyrinthe) et vous devez absolument passer par cette épreuve pour progresser. Et si vous n'en avez pas encore assez, pour chaque objet que vous voudrez vous procurer, il faudra passer par le dollarium (tremens ?).

Ce logiciel a l'avantage de présenter une aventure sur fond d'images digitalisées qui sont superbes ; par contre, le désir des auteurs d'insérer un peu d'action sous forme de deux petits jeux d'arcades n'est pas vraiment une réussite ; aussi, nous vous conseillons de faire une sauvegarde quand vous aurez réussi un passage de jeu d'arcade !





QUAD

MICROIDS
Arcade/Simulation

Avant de vous lancer (pur le rêve) dans le prochain Paris-Dakar, nous vous conseillons de bien vous installer devant votre écran de CPC, dans un fauteuil à siège bacquet, et de vous lancer dans une course folle à travers le désert au volant de votre Quad (qui n'est autre chose qu'un buggy avec quatre grosses roues...) !

A la simulation pure, Microids a ajouté à ce logiciel une partie arcade qui fait que vous devez résister à des attaques d'hélicoptères ou autres engins volants qui ne veulent pas que vous atteigniez le bout de la piste. C'est ainsi que non seulement vous devez éviter les nombreux obstacles de la piste (qui est très encombrée de cactus et autres... car peu utilisée) mais en plus vous devez savamment zigzaguer pour ne pas être atteint par les projectiles des avions...

et, pour couronner le tout, vous devez absolument éviter de sortir de la piste car alors vous ne quenez de vous enliser !

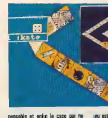
Avec ce logiciel, vous êtes confronté à une animation qui vous transporte à une vitesse folle !.. Ce qui fait que le Quad ne répond pas toujours comme vous le souhaitez mais peut être est ce dû à un manque d'entraînement... dans ce cas, Quad mérite le détour.



PLAY BAC
MICROIDS
Réflexion

Faisons une petite supposition : vous avez 18 ans, il est 8 h du matin et vous vous présentez à l'entrée d'une immense salle d'examen avec une multitude de petits bureaux à une place. Je suppose que cette vision vous rappelle assez bien les épreuves du bac. . Si vous étiez à nouveau confronté à cette situation, sortiriez-vous avec le diplôme à la main ?

Vous avez la possibilité de le savoir en vous amusant avec Play Bac ; en effet quatre matières sont à votre disposition : histoire, géographie, anglais et philosophie. Outre ces cases très sérieuses, vous avez en prime une case "détente" qui vous pose une question ou vous fait effectuer une action. Enfin, pour agrémenter le jeu, vous avez la case "bombe" où vos adversaires choisissent le thème de la question, la case "boum" où c'est vous qui faites votre choix, la case "bac blanc" qui constitue un test indispensable et enfin la case qui ne pouvait pas ne pas exister : la case "bonnet d'âne".



DETENTE
quel film de Robert Altman avec Sutherland et Elliott Gould, Pa d'or au Festival de Cannes 1970 déroule pendant la guerre de Co
VOTRE REPONSE ?

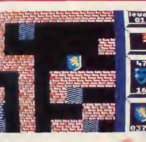
Comme vous l'avez sans doute déjà remarqué, Play Bac est un mélange des deux jeux suivants : un peu de jeu de l'oie et un peu de Trivial Pursuit. Comme dans ce dernier, la bonne réponse s'affiche à l'écran après que le joueur ait répondu et ce sont les adversaires qui décident de la validité de la réponse du joueur.. étant bien entendu que ce jeu est fait pour apprendre et non pour tricher...

Ce logiciel vous propose un bon moment de détente pouvant être vécu jusqu'à 6 joueurs , de plus la présentation en elle même est très agréable avec sa superposition de fenêtres... Seulement, il reste encore à savoir si une famille appréciera autant cette catégorie de jeux à l'écran que tous autour d'une table avec le jeu de société du même nom ?..

XOR
LOGOTRON
Arcade

XOR ! Cela ressemble à un cri de guerre mais, en fait, il s'agit d'une histoire de masques et de labyrinthes. Le but du jeu est très simple : vous devez découvrir la véritable nature de XOR et pour cela vous allez devoir reconstituer un indice en code formé de 15 lettres.

Pour y parvenir, vous n'avez pas moins de 15 labyrinthes à couvrir complètement car vous ne pouvez passer la porte de sortie et connaître une lettre qu'après avoir ramassé tous les masques présents dans chaque niveau. Au niveau 1, tout se passe à peu près bien au deuxième ou troisième essai car il faut considérer que le nombre de déplacements dans chaque labyrinthe est limité.



Mais à partir des niveaux suivants vont intervenir des poussins, des poissons ou même des bombes qui, suivant la manière dont ils sont manipulés peuvent bloquer les issues ou ouvrir des chemins..

Bref, ce premier jeu de Logotron ressemble à priori à un banal parcours de labyrinthes mais au fur et à mesure de votre progression, vous vous rendez compte que vous êtes obligés d'introduire un brin de stratégie. Le graphisme, par ailleurs, est assez enfantin mais suffisamment coloré pour présenter un intérêt à tous ceux qui veulent tester une certaine forme de logique.

INITIATION A CP/M

21

Francis VERSCHEURE

Nous terminons ce mois-ci notre étude des fonctions du BDOS de CP/M Plus.

FIXE LE MOT DE PASSE PAR DEFAUT

Lorsque l'on utilise le système de protection des fichiers par mots de passe pour accéder à un fichier il est nécessaire de préciser le mot de passe associé à ce fichier. Cette fonction permet donc d'indiquer le mot de passe qui doit être utilisé si le fichier auquel on veut accéder est protégé.
En entrée : C=106 DE = adresse d'une zone de 8 octets maximum qui contient le mot de passe.
En sortie : pas de code retour.

LECTURE DU NUMERO DE SERIE DE CP/M

Cette fonction permet de lire le numéro de série de votre disquette CP/M. Son utilisation permet donc d'instaurer une protection d'un programme en vérifiant que le CP/M sous lequel il tourne est le bon.
En entrée : C= 107 DE = pointe sur une zone de manœuvre de 3 octets. Le premier octet contiendra le numéro associé au constructeur du matériel et le mot suivant est le numéro de série lui-même sur 16 bits.
En sortie : pas de code retour, la zone de manœuvre contient le numéro de série.
Note importante : cette fonction a été modifiée par Amstrad et renvoie en fait 6 octets qui sont la chaîne '654321' en ASCII !

LECTURE/ECRITURE D'UN CODE RETOUR DE FIN DE PROGRAMME

Lorsque l'on enchaîne des programmes qui sont liés entre eux, comme par exemple une extraction de données d'un fichier suivi d'une édition, il est pratique de savoir dans le deuxième (ou énième) programme que le précédent s'est déroulé correctement. Ceci est possible par le mécanisme des codes retour. Ainsi, le programme positionne un code retour qui peut être lu par le programme suivant qui effectue ou non son traitement suivant la valeur de ce code.
En entrée : C=108 DE = 0FFFFH si on veut lire le code ou DE = la valeur du code.
En sortie : si on a lu le code il est contenu dans HL. Pas de code retour pour la fonction.

LECTURE/ECRITURE DU MODE CONSOLE

Cette fonction permet de connaître ou de positionner certains des paramètres de fonctionnement de la console.
En entrée : C=109 DE = 0FFFFH si on veut lire le mode console ou DE = le nouveau mode console.
En sortie : mode console dans HL s'il s'agit d'une lecture.
Voici la définition des bits du mode console :
— Bit 0 : 1= la fonction 11 de lecture de l'état de la console ne prend pas en compte la frappe d'un CTRL-C.
0= la fonction 11 prend en compte un CTRL-C
— Bit 1 : 1= supprime l'action de CTRL-S CTRL-Q.
0= autorise l'action de CTRL-S CTRL-Q.
— Bit 2 : 1= supprime la reconnaissance des TAB et l'action du CTRL-P.
0= remplace les TAB par des blancs modulo 8 et autorise l'action du CTRL-P.
— Bit 3 : 1= ignore les CTRL-C.
0= prend en compte les CTRL-C

LECTURE/ECRITURE DU DELIMITEUR DE CHAINE

La fonction 9 qui écrit une chaîne de caractères à l'écran utilise un délimiteur de fin de chaîne qui par défaut est le $.
Cette fonction permet donc de modifier la valeur de ce délimiteur.
En entrée : C=110 DE = 0FFFFH si on veut lire le délimiteur ou la valeur du nouveau délimiteur dans E.
En sortie : si on lit le délimiteur celui-ci est dans l'accumulateur A.

AFFICHAGE D'UN BLOC MEMOIRE

Cette fonction permet d'afficher à l'écran une zone quelconque de mémoire. Mais attention, elle ne peut être utilisée que pour des codes ASCII visualisables, c'est donc en quelque sorte une variante de la fonction 9, qui permet d'afficher une chaîne terminée par un délimiteur. Certains codes de contrôle comme les Line Feed ou Carriage Return sont pris en compte.
En entrée : C=111 DE = adresse de 2 mots de 16 bits. Le premier contient l'adresse début de la zone et le second la taille.
En sortie : pas de code retour.

IMPRESSION D'UN BLOC MEMOIRE

Identique à la fonction précédente, mais imprime au lieu de visualiser.
En entrée : C=112 DE = adresse de 2 mots : - adresse de la zone + taille
En sortie : pas de code retour.

CONSTRUCTION D'UN FCB A PARTIR D'UNE CHAINE DE CARACTERES

Cette fonction permet de construire un FCB prêt à être utilisé, à partir d'une chaîne de caractères contenant le nom d'un fichier précédé ou non d'une lettre de disque. Cette chaîne doit être terminée par un octet à 0.
En entrée : C = 152 DE = adresse de 2 mots. Le premier est l'adresse de la chaîne à convertir (128 octets max.) et le second l'adresse d'une zone où construire le FCB (36 octets mini)
En sortie : HL = 0 si la construction est OK et un pointeur sur le caractère en erreur dans la chaîne dans le cas contraire.

Voila, nous sommes arrivés au bout de notre étude et maintenant vous en savez assez pour écrire de nombreux programmes tout en assembleur. Certes l'écriture en assembleur est plus complexe et plus technique que l'écriture en BASIC ou en PASCAl, mais le jeu en vaut la peine car le gain de vitesse à l'exécution est important ainsi que le gain de temps lors des mises au point. En effet avec un bon 'débugger' lorsque le programme est écrit en assembleur on voit tout de suite ou doivent être les choses, ce qui n'est pas le cas avec le code résultant d'une compilation !
Bref, chacun voit midi à sa porte et toutes les positions sont respectables, alors pour ceux qui aiment bidouiller les octets, l'assembleur reste le meilleur outil disponible et en plus sous CP/M Plus c'est gratuit !

QUELQUES DERNIERS EXEMPLES :

```
BDOS     equ      5

NUMCPM   equ      107
MODECONS equ      109
DELIM    equ      110
BPRINT   equ      111

; Exemple de lecture du numero de serie a la 'Amstrad'

         mvi      c,NUMCPM
         lxi      d,NUMSERIE
         call     BDOS
         ...
         ...
NUMSERIE:         db       '123456'          ; Zone de 6 octets

; Exemple de lecture du mode console

         mvi      c,MODECONS                 ; Lecture du mode console
         lxi      d,-1
         call     BDOS                       ; le resultat est dans HL
         shld     OLDCONSMODE
         ...
         ...
OLDCONSMODE:      dw       0

; Exemple de lecture du delimiteur utilise par la fonction 9

         mvi      c,DELIM               ; Lecture du delimiteur
         lxi      d,-1
         call     BDOS                  ; le resultat est dans A.
         sta      OLDDELIM
         ...
         ...
OLDDELIM:         db       0

; Exemple d'utilisation de la fonction d'affichage de bloc memoire

         mvi      c,BPRINT
         lxi      d,CCB
         call     BDOS
         ...
         ...
CCB:     dw       DEBUT
         dw       FIN-DEBUT         ; La taille de la zone a afficher

DEBUT:   db       'Voici un texte a afficher d''un seul coup '',0AH,0DH
         db       'Ceci doit se trouver sur la deuxieme ligne.'
FIN:     db       0
```